IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Application of: Dwip N. Banerjee	:
Serial No: 10/777,715	: Before the Examiner: : Igor V. Chernyak
Filed: 02/12/2004	: Group Art Unit: 2183
Title: SYSTEM, APPARATUS AND METHOD OF AGGREGATING TCP-OFFLOADED ADAPTERS	: Confirmation No.: 5927

APPELLANTS' BRIEF UNDER 37 C.F.R. §41.37

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal to a final rejection dated April 30, 2008 of the claims in the Application. This brief is submitted pursuant to a Notice of Appeal filed on July 28, 2008 in accordance with 37 C.F.R. §41.31.

BRIEF FOR APPLICANTS - APPELLANTS

(i)

Real Party in Interest

The real party in interest is International Business Machines Corporation (IBM), the assignee.

(ii)

Related Appeals and Interferences

There are no other appeals or interferences known to appellants, appellants' representative or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii)

Status of Claims

Claims 1 – 20 are finally rejected. Claims 1 – 20 are being appealed.

(iv)

Status of Amendment

An "Amendment-After-Final" was not filed.

(v)

Summary of Claimed Subject Matter

The invention, as claimed in Claim 1, provides a method of aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters of a first communications system to augment network data transaction bandwidth of the first communications system by a factor of N, N being an integer. The method comprises aggregating the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters (support is on page 4, lines 4 – 7), selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and a second communications

systems is to originate (support is on page 15, lines 18 – 21), originating the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the first and the second communications systems (support is on page 16, lines 15 – 21); and transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter (support is on page 11, lines 3 – 8 and on page 16, lines 8 – 9 as well as switch 430 of Fig. 4).

The invention, as claimed in Claim 6, provides a computer program product on a computer readable medium for aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters of a first communications system to augment network data transaction bandwidth of the first communications system by a factor of N, N being an integer. The computer program product comprises: code means for aggregating the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters (support is on page 4, lines 4 – 7); code means for selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and a second communications systems is to originate (support is on page 15, lines 18 – 21); code means for originating the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the first and the second communications systems (support is on page 16, lines 15 – 21); and code means for transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter (support is on page 11, lines 3 – 8 and on page 16, lines 8 – 9 as well as switch 430 of Fig. 4). The code means of the invention are the processes whose steps are delineated in Figs. 6 and 7 and on page 16, line 22 to page 17, line 26.

The invention, as claimed in Claim 11, provides an apparatus for aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters of a first communications system to augment network data transaction bandwidth of the first communications system by a factor of N, N being an integer. The apparatus comprises: means for aggregating the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters (support is on page 4, lines 4 – 7); means for selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and a second communications systems is to originate (support is on page 15, lines 18 – 21); means for originating the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the first and the second communications systems (support is on page 16, lines 15 – 21); and means for transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter (support is on page 11, lines 3 – 8 and on page 16, lines 8 – 9 as well as switch 430 of Fig. 4). The means of the invention include any and all of the processors 202, 204 of Fig. 2 and 302 of Fig. 3 processing the code means whose steps are delineated in Figs. 6 and 7 and on page 16, line 22 to page 17, line 26.

The invention, as claimed in Claim 16, provides a system for aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters to augment network data transaction bandwidth by a factor of N, N being an integer. The system comprises: at least one storage device (local memory 209, hard disk 232 of Fig. 2, main memory 304, memory 324, disk 325, tape 328 of Fig. 3) for storing code data; and at least one processor (processors 202, 204 of Fig. 2 and 302 of Fig. 3) for processing the code data to aggregate the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters (support is on page 4, lines 4 – 7), to select one of the aggregated N TCP-offloaded adapters through which a connection between the system and a

remote communications system is to originate (support is on page 15, lines 18 – 21), to originate the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the system and the remote communications systems (support is on page 16, lines 15 – 21), and to transact data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter (support is on page 11, lines 3 – 8 and on page 16, lines 8 – 9 as well as switch 430 of Fig. 4).

(vi)

Grounds of Rejection to be Reviewed on Appeal

Whether it was proper to reject Claims 1 - 20 under 35 USC §102(e) as being anticipated by Vangal et al.

(vii)

Arguments

Claims 1, 6, 11 and 16

Vangal et al, purport to teach techniques for coordinating operation of multiple network protocol off-load engines (e.g., Transport Control Protocol (TCP) off-load engines). According to the teachings of Vangal et al., a scheme that aggregates multiple off-load engines is provided. In the scheme, a controller is used to distribute responsibility for handling different network connections across the different engines. When a packet arrives that identifies a connection not previously seen, the controller allocates an engine for handling the packet. An interface is also used to receive network data by the engine which, once processed by an engine, is sent to a host system.

The Examiner used Fig. 1 of Vangal et al. to show that Vangal et al. teach transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data

associated with the connection from the network to the selected TCP-offloaded adapter. Appellants respectfully disagree.

Fig. 1 of Vangal et al. shows a controller 102 connected to N (TCP) off-load engines 100. Each N (TCP) off-load engine 100 is connected to an input bus 106 for receiving data and an output bus 104 for transferring the data to a host computer system. The controller 102 coordinates operation of the engines 100. For example, when a packet arrives, each one of the N (TCP) off-load engines 100 checks to see whether it is the one that originally established the connection. The (TCP) off-load engine 100 that originally established the connection then handles the packet. If none of (TCP) off-load engines 100 established the connection (i.e., if it is a packet that is being used to establish a new connection), the controller 102 allocates an engine 100 to handle the connection. When a (TCP) off-load engine 100 is processing packets of data, the controller 102 limits the use of the output bus 104 to the engine 100 processing the data. Further, the controller 102 may also selectively enable and disable different engines 100 under different circumstances to save power and decrease heat generated by the collection of engines 100.

However, Vangal et al. do not teach, show or suggest a computing device that assembles data from the N TCP-offloaded adapters to send to a network.

Further, Vangal et al. do not teach, show or suggest the computing device channeling data associated with a connection from the network to a selected TCP-offloaded adapter.

It is a well settled law that in considering a Section 102 rejection, all the elements of the claimed invention must be disclosed in a single item of prior art in the form literally defined in the claim. *Jamesbury Corp. v. Litton Indus. Products*, 756 F.2d 1556, 225 USPQ 253 (Fed. Cir. 1985); *Atlas Powder Co. v. Dupont*, 750 F.2d 1569, 224 USPQ 409 (Fed. Cir. 1984); *American Hospital Supply v. Travenol Labs.*, 745 F.2d 1, 223 USPQ 577 (Fed. Cir. 1984).

In this particular case, Vangal et al. do not teach, show or suggest ***transacting data through a computing device, the computing device to***

assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter as asserted by the Examiner.

Appellants submit that Claims 1, 6, 11 and 16 are not anticipated by Vangal et al.

Claims 2, 7, 12 and 17

Claims 2, 7, 12 and 17 include the limitations "wherein selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and the second communications systems is to originate is based on a local port and a remote port, the local port and the remote port being the ports through which the data transaction is to occur."

The Examiner asserted that Vangal et al. teach those limitations in a combination of paragraphs (i.e., paragraphs [0027], [0028], [0024], [0038] and [0046]). Appellants respectfully disagree.

In paragraph [0027], Vangal et al. disclose a scheme for aggregating n-off-load engines 100a-100n allowing C-connections to be supported by a single engine such that n-aggregated engines 100a-100n can support n.times.C connections. This scheme is further explained in paragraph [0028].

In paragraph [0024], Vangal et al. explain the reason behind using the n-aggregated off-load engines.

In paragraph [0038], Vangal et al. disclose:

> [0038] FIG. 4 depicts a flow chart of a process for allocating connections to different off-load engines. As shown, after receiving 140 a packet, the process determines 142 whether the packet is part of a new connection or is part of one already allocated to an engine. For example, a connection may be defined as a combination of a packet's Internet Protocol (IP) source and destination addresses, transport layer protocol, and source and destination ports. An engine may lookup a packet's connection data

> within data identifying connections allocated to the engine. If an engine reports a "hit", the engine 152 may process the packet (e.g., perform TCP operations). Otherwise, the controller may allocate an engine for the new connection.

Thus, in that paragraph Vangal et al. disclose a flowchart of a process that is used when receiving data. Specifically, when data is received, it is determined whether the data is associated with an existing connection. If so, the off-load engine that originated the connection will process the data. If, however, the data is not associated with an existing connection (i.e., if it is data for establishing a new connection), the controller will allocate one of the off-load engines to originate the new connection.

In order to determine whether the data is associated with an existing connection, IP source and destination addresses, transport layer protocol, and source and destination ports are used. That is, the n-off-load engines compare the IP source and destination addresses, transport layer protocol, and source and destination ports in the received data with IP source and destination addresses, transport layer protocol, and source and destination ports of all existing connections that they are handling. If one of the n-off-load engines finds a match then it is the one that originated the connection and therefore it is the one that will process the data. As mentioned above, if none the of the n-off-load engines finds a match, the controller will allocate one of the off-load engine to originate the new connection.

The method by which the comparison is made is explained in paragraph [0046].

In paragraph [0032], reproduced below in its entirety, Vangal et al. specifically disclose a method by which the controller selects one of the n-off-load engines to establish a new connection.

> [0032] Potentially, in the case of a packet signaling the start of a new connection, no engine 100 will signal a "hit" for the packet. Thus, the controller 102 allocates the connection to one of the

> engines 100. The controller 102 may implement an allocation scheme based on current engine 100 usage. To provide the controller 102 with information about the engine's 100 current usage, the engine 100 can output a "full" signal 110 that identifies when the engine 100 cannot handle additional connections. In response to a "full" signal 10, the controller 102 may select a different engine to use when a new connection needs to be allocated.

Thus, in that method, source and destination ports are not considered at all. Rather, the controller takes into consideration whether a particular off-load engine can handle a new connection to determine whether to select that off-load engine to originate the new connection (see also paragraphs [0039] and [0040]).

However, Vangal et al. do not teach, show or suggest in any one of the cited paragraphs or in a combination thereof the limitations ***wherein selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and the second communications systems is to originate is based on a local port and a remote port, the local port and the remote port being the ports through which the data transaction is to occur*** as asserted by the Examiner.

Hence, Appellants submit that Claims 2, 7, 12 and 17 are not anticipated by Vangal et al.

Claims 3, 8, 13 and 18

Claims 3, 8, 13 and 18 include the limitations "wherein selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and the second communications systems is to originate includes the step of assigning a local port through which the connection is to occur if a local port was not yet assigned."

The Examiner asserted that Vangal et al. teach the limitations "assigning a local port through which the connection is to occur if a local port was not yet

assigned" in paragraphs [0038], [0039], [0046], [0047] and [0054] – [0056]. Appellants respectfully disagree.

As mentioned above, in paragraph [0038], Vangal et al. disclose a flowchart of a process that is used when receiving data. Specifically, when data is received, it is determined whether the data is associated with an existing connection. If so, the off-load engine that originated the connection will process the data. If, however, the data is not associated with an existing connection (i.e., it is data for establishing a new connection), the controller will allocate one of the off-load engines to originate the new connection.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the connection is to occur if a local port was not yet assigned*** as asserted by the Examiner.

In paragraph [0039], Vangal et al. disclose a couple methods that the controller may use to allocate one of the off-load engines to originate the connection. For example, Vangal et al. disclose that a round-robin scheme may be used to allocate the off-load engines or the controller may want to concentrate the connections on one engine and thus use that engine to originate the connection so long as the engine can handle one more connection.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the connection is to occur if a local port was not yet assigned*** as asserted by the Examiner.

As explained above, in paragraph [0046] Vangal disclose the comparison method used by the n-off-load engines to determine whether any one of them had originated a connection and thus should process an arriving piece of data associated with that connection.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the***

connection is to occur if a local port was not yet assigned as asserted by the Examiner.

In paragraph [0047], Vangal disclose:

> [0047] When an engine is allocated a new connection by the controller, the current packet may represent the start of a new connection. Thus, the engine 100 can initialize the working register 168 (e.g., set to the "LISTEN" state in TCP) and allocate CAM 164 and context data 166 entries for the connection, for example, using a LRU (Least Recently Used) algorithm or other allocation scheme.

In that paragraph, Vangal et al. explain that when an engine has been selected to handle a new connection, the engine will store data pertaining to that connection (e.g., IP source and destination addresses, transport layer protocol, and source and destination ports) into a content addressable memory (CAM). This facilitates future comparisons to determine whether an arriving piece of data (associated with that connection) should be processed by the engine.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the connection is to occur if a local port was not yet assigned*** as asserted by the Examiner.

In paragraphs [0054] – [0056], Vangal et al. disclose:

> [0054] In a system aggregating a collection of the sample engine illustrated in FIG. 5, engine 100 CAMs are initially "empty" and no engine receives a "grant" signal. After receiving a packet forming part of a connection, the controller 102 selects an engine 100 for allocation of the connection and asserts the grant signal 116 to the selected engine 100. The selected engine 100 processes the incoming segment and updates its CAM 164 and TCB 166

> contents. The grant 116 also enables the engine 100 to drive data on the bus 104 for the duration of the segment.
>
> [0055] The controller 102 can allocate further connections to the engine 100 until the engine 100 signals "full" 110. The "full" signal 110 causes the controller 102 to allocate new connections to a different engine (though potentially an already active engine).
>
> [0056] A newly arriving segment could be either an existing connection allocated to one of the active engines or an unallocated (e.g., new) connection. To determine whether an engine 100 has already been allocated for the packet's connection, the controller 102 asserts the grant signal to engines 100 currently servicing at least one on-going connection. These engines 100 perform a connection (e.g., CAM 164) lookup in parallel. For a new connection (e.g., no engine reports a CAM 164 hit), the controller 102 allocates the connection to the most recently activated engine 100, provided it is not signaling "full" 110. If a "full" signal 110 is asserted by the engine 100, however, the controller then picks the next most recently activated engine, for example, by accessing a queue identifying engines most recently allocated their first connection. This process may continue until an active engine 100 is identified that is not asserting a "full" signal 110. If no such active engine 100 is identified, the controller 102 may activate another engine 100 and allocate the connection to it.

Thus, in paragraph [0054], Vangal et al. explain the processing method used by an engine that has been selected to process an arriving piece of data.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the***

connection is to occur if a local port was not yet assigned as asserted by the Examiner.

In paragraph [0055], Vangal et al. explain that the controller can assign new connections to one engine until the engine cannot handle anymore new connections.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the connection is to occur if a local port was not yet assigned*** as asserted by the Examiner.

In paragraph [0056], Vangal et al. explain that the scheme used by the controller and the engines to determine which engine should handle an arriving piece of data.

However, Vangal et al. do not teach, show or so much as suggest in that paragraph the limitations of ***assigning a local port through which the connection is to occur if a local port was not yet assigned*** as asserted by the Examiner.

Hence, Appellants submit that Claims 3, 8, 13 and 18 are not anticipated by Vangal et al.

Claims 4, 9, 14 and 19

Claims 4, 9, 14 and 19 include the limitations "wherein the assigned local port is an ephemeral port."

The Examiner asserted that Vangal et al. teach those limitations in paragraphs [0038], [0039], [0046], [0047] and [0054] – [0056]. Appellants respectfully disagree.

As mentioned above, in paragraph [0038], Vangal et al. disclose a flowchart of a process that is used when receiving data. Specifically, when data is received, it is determined whether the data is associated with an existing connection. If so, the off-load engine that originated the connection will process the data. If, however, the data is not associated with an existing connection (i.e.,

it is data for establishing a new connection), the controller will allocate one of the off-load engines to originate the new connection.

However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

In paragraph [0039], Vangal et al. disclose a couple methods that the controller may use to allocate one of the off-load engines to originate the connection. For example, Vangal et al. disclose that a round-robin scheme may be used to allocate the off-load engines or the controller may want to concentrate the connections on one engine and thus use that engine to originate the connection so long as the engine can handle one more connection.

However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

In paragraph [0046] Vangal disclose the comparison method used by the n-off-load engines to determine whether any one of them had originated a connection and thus should process an arriving piece of data associated with that connection.

However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

In paragraph [0047], Vangal et al. explain that when an engine has been selected to handle a new connection, the engine will store data pertaining to that connection (e.g., IP source and destination addresses, transport layer protocol, and source and destination ports) into a content addressable memory (CAM). This facilitates future comparisons to determine whether an arriving piece of data (associated with that connection) should be processed by the engine.

However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

In paragraph [0054], Vangal et al. explain the processing method used by an engine that has been selected to process an arriving piece of data.
However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

In paragraph [0055], Vangal et al. explain that the controller can assign new connections to one engine until the engine cannot handle anymore new connections.

However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

In paragraph [0056], Vangal et al. explain that the scheme used by the controller and the engines to determine which engine should handle an arriving piece of data.

However, Vangal et al. do not teach, show or suggest the limitations ***wherein the assigned local port is an ephemeral port*** in that paragraph as asserted by the Examiner.

Hence Appellants submit that Claims 4, 9, 14 and 19 are not anticipated by Vangal et al.

For the above-mentioned reasons, Appellants request reversal of the rejection and passage to issue of the claims.

Respectfully Submitted

By: _____
Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

(viii)

Claims Appendix

1. (Previously presented) A method of aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters of a first communications system to augment network data transaction bandwidth of the first communications system by a factor of N, N being an integer, the method comprising:

aggregating the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters;

selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and a second communications systems is to originate;

originating the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the first and the second communications systems; and

transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter.

2. (Previously presented) The method of Claim 1 wherein selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and the second communications systems is to originate is based on a local port and a remote port, the local port and the remote port being the ports through which the data transaction is to occur.

3. (Previously presented) The method of Claim 2 wherein selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and the second communications systems is to originate includes the step of assigning a local port through which the connection is to occur if a local port was not yet assigned.

4. (Original) The method of Claim 3 wherein the assigned local port is an ephemeral port.

5. (Previously presented) The method of Claim 1 wherein the data includes incoming and outgoing data, the incoming data being divided into data packets, each packet having associated therewith a local port and a remote port for selecting a TCP-offloaded adapter through which to be channeled.

6. (Previously presented) A computer program product on a computer readable medium for aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters of a first communications system to augment network data transaction bandwidth of the first communications system by a factor of N, N being an integer, the computer program product comprising:

code means for aggregating the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters;

code means for selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and a second communications systems is to originate;

code means for originating the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the first and the second communications systems; and

code means for transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter.

7. (Original) The computer program product of Claim 6 wherein the selecting code means includes code means for using a local port and a remote port to select the TCP-offloaded, the local port and the remote port being the ports through which the data transaction is to occur.

8. (Original) The computer program product of Claim 7 wherein the selecting code means includes code means for assigning a local port through which the connection is to occur if a local port was not yet assigned.

9. (Original) The computer program product of Claim 8 wherein the assigned local port is an ephemeral port.

10. (Previously presented) The computer program product of Claim 6 wherein the data includes incoming and outgoing data, the incoming data being divided into data packets, each packet having associated therewith a local port and a remote port for selecting a TCP-offloaded adapter through which to be channeled.

11. (Previously presented) An apparatus for aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters of a first communications system to augment network data transaction bandwidth of the first

communications system by a factor of N, N being an integer, the apparatus comprising:

means for aggregating the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters;

means for selecting one of the N aggregated TCP-offloaded adapters through which a connection between the first and a second communications systems is to originate;

means for originating the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the first and the second communications systems; and

means for transacting data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter.

12. (Original) The apparatus of Claim 11 wherein the selecting means includes means for using a local port and a remote port to select the TCP-offloaded, the local port and the remote port being the ports through which the data transaction is to occur.

13. (Original) The apparatus of Claim 12 wherein the selecting means includes means for assigning a local port through which the connection is to occur if a local port was not yet assigned.

14. (Original) The apparatus of Claim 13 wherein the assigned local port is an ephemeral port.

15. (Previously presented) The apparatus of Claim 11 wherein the data includes incoming and outgoing data, the incoming data being divided into data packets, each packet having associated therewith a local port and a remote port for selecting a TCP-offloaded adapter through which to be channeled.

16. (Previously presented) A system for aggregating N Transport Control Protocol-offloaded (TCP-offloaded) adapters to augment network data transaction bandwidth by a factor of N, N being an integer, the system comprising:

at least one storage device for storing code data; and

at least one processor for processing the code data to aggregate the N TCP-offloaded adapters by assigning a common Internet Protocol (IP) address to the N TCP-offloaded adapters, to select one of the aggregated N TCP-offloaded adapters through which a connection between the system and a remote communications system is to originate, to originate the connection using the selected TCP-offloaded adapter, the connection for transacting data over a network between the system and the remote communications systems, and to transact data through a computing device, the computing device to assemble data from the N TCP-offloaded adapters to the network and for channeling data associated with the connection from the network to the selected TCP-offloaded adapter.

17. (Original) The system of Claim 16 wherein processing the code data to select one of the TCP-offloaded adapter includes processing the code data to use a local port and a remote port to select the TCP-offloaded, the

local port and the remote port being the ports through which the data transaction is to occur.

18. (Original) The system of Claim 17 wherein the code data to select one of the TCP-offloaded adapter includes processing the code data to assign a local port through which the connection is to occur if a local port was not yet assigned.

19. (Original) The system of Claim 18 wherein the assigned local port is an ephemeral port.

20. (Previously presented) The system of Claim 16 wherein the data includes incoming and outgoing data, the incoming data being divided into data packets, each packet having associated therewith a local port and a remote port for selecting a TCP-offloaded adapter through which to be channeled.

(ix)

Evidence Appendix

None.

(x)

Related Proceedings Appendix

None.